



ARL-CR-0773 • MAY 2015



US Army Research Laboratory

# **Demonstration of Supervisory Control and Data Acquisition (SCADA) Virtualization Capability in the US Army Research Laboratory (ARL)/Sustaining Base Network Assurance Branch (SBNAB) US Army Cyber Analytics Laboratory (ACAL) SCADA Hardware Testbed**

**prepared by Daniel T Sullivan**  
*Raytheon Company*  
*22260 Pacific Blvd*  
*Dulles, VA*

*and*

**Edward J Colbert, PhD**  
*ICF International*  
*7125 Thomas Edison Drive #100*  
*Columbia, MD*

**under contract W911QX-14-F-0020**

Approved for public release; distribution unlimited.

## **NOTICES**

### **Disclaimers**

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.



**Demonstration of Supervisory Control and Data Acquisition (SCADA) Virtualization Capability in the US Army Research Laboratory (ARL)/Sustaining Base Network Assurance Branch (SBNAB) US Army Cyber Analytics Laboratory (ACAL) SCADA Hardware Testbed**

**prepared by Daniel T Sullivan**  
*Raytheon Company*  
*22260 Pacific Blvd*  
*Dulles, VA*

*and*

**Edward J Colbert, PhD**  
*ICF International*  
*7125 Thomas Edison Drive #100*  
*Columbia, MD*

**under contract W911QX-14-F-0020**

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 0704-0188	
<p>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p><b>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</b></p>					
1. REPORT DATE (DD-MM-YYYY) May 2015		2. REPORT TYPE Final		3. DATES COVERED (From - To) 07/2014–12/2014	
4. TITLE AND SUBTITLE Demonstration of Supervisory Control and Data Acquisition (SCADA) Virtualization Capability in the US Army Research Laboratory (ARL)/Sustaining Base Network Assurance Branch (SBNAB) US Army Cyber Analytics Laboratory (ACAL) SCADA Hardware Testbed				5a. CONTRACT NUMBER W911QX-14-F-0020	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Daniel T Sullivan and Edward J Colbert, Ph.D				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Raytheon Company ICF International 22260 Pacific Blvd 7125 Thomas Edison Drive #100 Dulles, VA 20166 Columbia, MD 21046				8. PERFORMING ORGANIZATION REPORT NUMBER ARL-CR-0773	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) US Army Research Laboratory ATTN: RDRL-CIN-S 2800 Powder Mill Road Adelphi, MD 20783-1138				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.					
13. SUPPLEMENTARY NOTES ARL POC: Robert J Reschly					
14. ABSTRACT In support of the US Army Research Laboratory (ARL) mission to conduct cybersecurity research to protect Industrial Control Systems (ICS), the ARL Sustaining Base Network Assurance Branch (SBNAB) constructed a Supervisory Control and Data Acquisition (SCADA) hardware testbed to simulate the network traffic between human machine interface (HMI) and programmable logic controller (PLC) components. The HMI and PLC components were instantiated with software and installed in multiple virtual machines (VMs) to emulate 6 conceptual manufacturing plant processes. Two experiments were conducted: <ul style="list-style-type: none"> <li>• Validate the virtualized network performance by creating and capturing HMI–PLC network traffic over a 24-h period in the virtualized network and inspect the packets for errors.</li> <li>• Test the interoperability of physical network elements with the virtualized network. In this test, a simulated threat actor used a laptop computer to connect to the virtualized production network and send malicious Modbus network commands to create a manipulation of view attack.</li> </ul> The results of both experiments are PASS. The experiments validated the capability to establish a SCADA hardware testbed using virtualization and this infrastructure is now part of the ARL SBNAB US Army Cyber Analytics Laboratory (ACAL).					
15. SUBJECT TERMS SCADA, Modbus, virtualization					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 32	19a. NAME OF RESPONSIBLE PERSON Daniel T Sullivan
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) 301-394-0248

## Contents

---

<b>List of Figures</b>	<b>iv</b>
<b>List of Tables</b>	<b>iv</b>
<b>Acknowledgments</b>	<b>v</b>
<b>1. Background and Motivation</b>	<b>1</b>
<b>2. Description of Test</b>	<b>1</b>
2.1 Test Processes	1
2.2 Virtual Representation of the MRE SCADA system	2
2.3 PLC Configuration	4
2.4 HMI Configuration	5
<b>3. Execution of MRE Test</b>	<b>9</b>
3.1 Network Virtualization Subtest	9
3.2 Simulated Cyber Attack	10
<b>4. MRE Test Results</b>	<b>11</b>
4.1 Network Virtualization Subtest	11
4.2 Simulated Cyber Attack	11
<b>5. Conclusions</b>	<b>13</b>
<b>6. References</b>	<b>14</b>
<b>Appendix A. Experiment Hardware and Software</b>	<b>15</b>
<b>Appendix B. ModbusPal Tables</b>	<b>17</b>
<b>List of Symbols, Abbreviations, and Acronyms</b>	<b>23</b>
<b>Distribution List</b>	<b>24</b>

## List of Figures

Fig. 1	Process map for MRE SCADA system .....	2
Fig. 2	Testbed architecture .....	3
Fig. 3	Field network VMs .....	4
Fig. 4	Overall plant HMI dashboard .....	6
Fig. 5	Chicken cooker dashboard .....	7
Fig. 6	Vegetable cooker dashboard .....	7
Fig. 7	Meal preparation dashboard .....	8
Fig. 8	High-pressure processing dashboard .....	8
Fig. 9	Main conveyor belt dashboard .....	9
Fig. 10	Product packaging dashboard .....	9
Fig. 11	Meal preparation dashboard before cyber attack .....	12
Fig. 12	Meal preparation dashboard after cyber attack .....	12
Fig. 13	Meal preparation alarm panel after cyber attack .....	13

## List of Tables

Table	Network virtualization test results .....	11
Table A-1	Hardware list .....	16
Table A-2	Software list .....	16
Table B-1	Configuration and measurements for chicken cooker PLC .....	18
Table B-2	Configuration and measurements for vegetable cooker PLC .....	19
Table B-3	Configuration and measurements for meal preparation and packaging PLC .....	20
Table B-4	Configuration and measurements for high-pressure processing PLC .....	21
Table B-5	Configuration and measurements for main conveyor belt PLC .....	22
Table B-6	Configuration and measurements for packaging PLC .....	22

## Acknowledgments

---

We greatly appreciate Dr Alexander Kott, Curtis Arnold, and Chuck Smith for supporting Industrial Control Systems/Supervisory Control and Data Acquisition (ICS/SCADA) research at the US Army Research Laboratory (ARL). We are grateful to Kin Wong and Carlos Mateo for help with the SCADA lab design and ordering lab equipment. Max Turk, Akhil Oniha, and James Herron were very helpful in setting up hardware and software in the testbed.

INTENTIONALLY LEFT BLANK.



## **1. Background and Motivation**

---

This report describes a test experiment executed on the US Army Research Laboratory (ARL) Sustaining Base Network Assurance Branch (SBNAB) Supervisory Control and Data Acquisition (SCADA) hardware testbed. This initial test experiment has been executed to demonstrate SCADA virtualization capability on the testbed. The SCADA hardware testbed is part of the US Army Cyber Analytics Laboratory (ACAL), which provides hardware and network infrastructure and other support needed for collaboration between ARL and other government and commercial institutions.

In this test, we use a software-emulated programmable logic controller (PLC) and public domain human machine interface (HMI) controller software instead of actual PLC hardware and vendor-based HMI software. Both PLC and HMI controller software run inside virtual machines (VMs), allowing the entire SCADA system to be virtualized. In the future, real PLC hardware and commercial HMI software will also be used in ACAL SCADA testbed research experiments.

This initial test of the ACAL SCADA testbed emulates network traffic found in SCADA systems (or Industrial Control Systems [ICS]), as we demonstrate below.

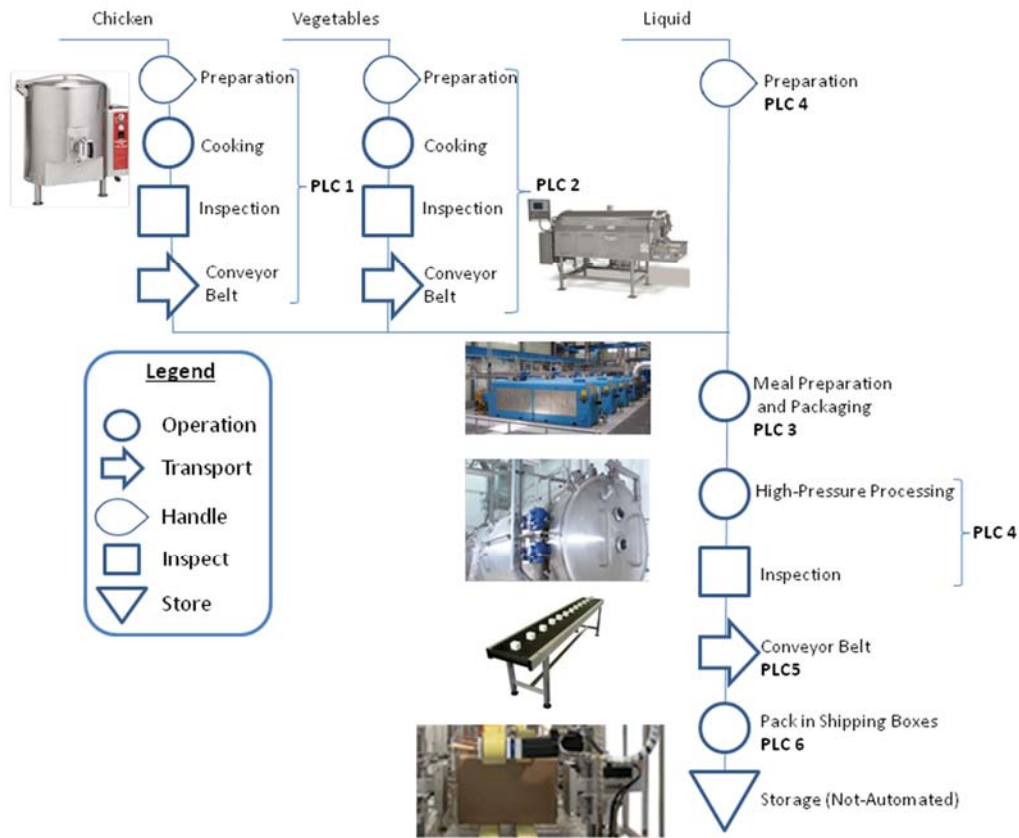
## **2. Description of Test**

---

### **2.1 Test Processes**

---

The SCADA system emulated in this test is that of a conceptual Meals-Ready-to-Eat (MRE) manufacturing process. The process map for the system is illustrated in Fig. 1 and shows 6 PLCs controlling various pieces of machinery used to produce the MREs.

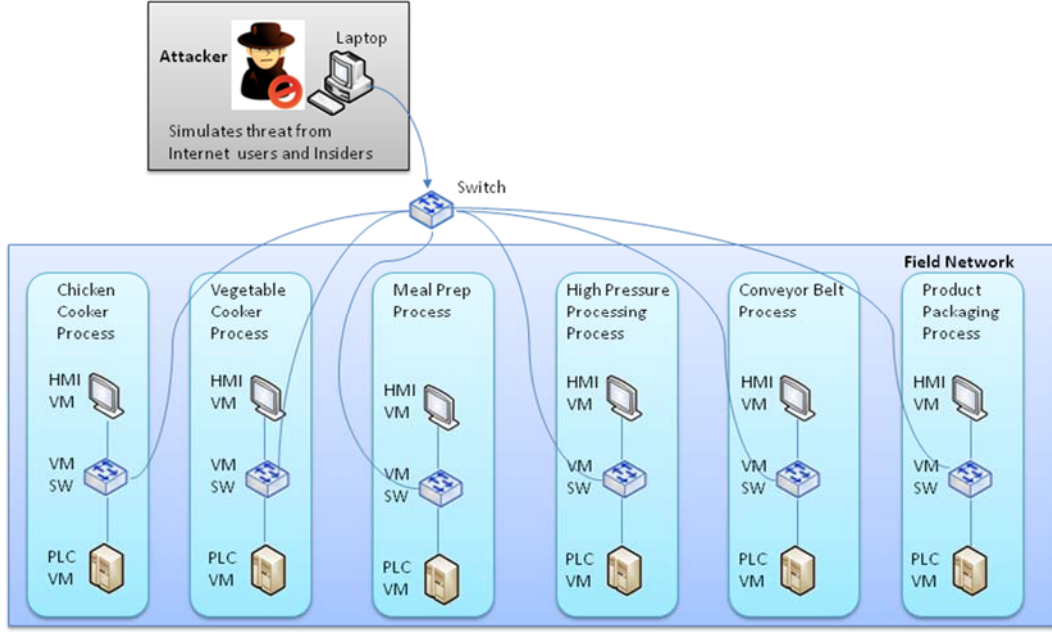


**Fig. 1 Process map for MRE SCADA system**

## 2.2 Virtual Representation of the MRE SCADA system

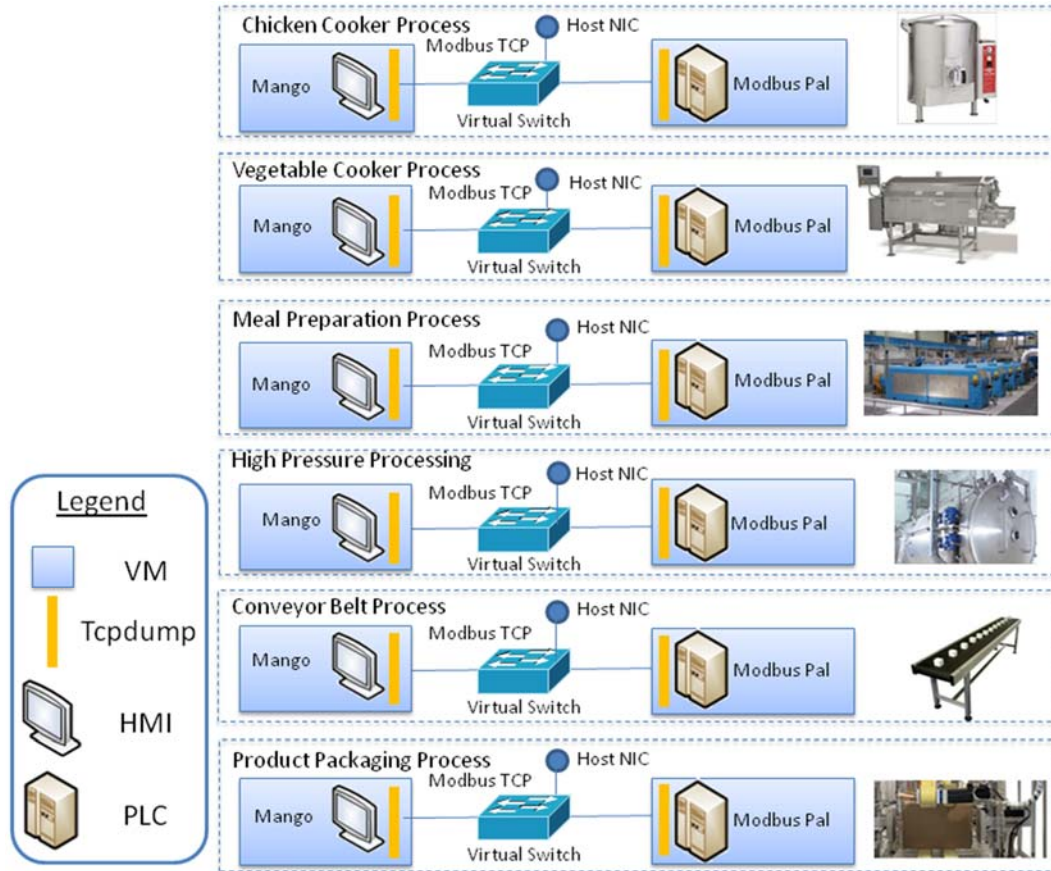
The software used in the test emulates the traffic sent and received by PLCs and HMIs found in MRE SCADA processes. The PLCs control machinery and receive sensor inputs from physical plant components. HMIs are computers running control software that frequently polls a PLC for status information about the controlled process. A human plant operator monitors the HMI computer and software. HMIs may also provide a capability for the human operator to manually control a process, if needed.

In this experiment, the HMI and PLC components function within VMs. The testbed topology of VMs used for the MRE SCADA test is depicted in Fig. 2. Six pairs of PLCs and HMIs have been constructed inside a virtual network, and all 12 VMs are connected to virtual switches. An attacker, who also has access to the virtual network via a virtual switch, can initiate attacks on the MRE SCADA system.



**Fig. 2 Testbed architecture**

A more detailed diagram of the simulated SCADA network is shown in Fig. 3 and additional information is presented in Appendix A. Experiment Hardware and Software. In this experiment, each HMI polls a simulated PLC using the industrial Modbus transmission control protocol (TCP). The HMI software used in this test is the open source Mango Automation application,<sup>1</sup> while the simulated PLC software is the open source ModbusPal Java application. When queried using the Modbus TCP protocol, ModbusPal reports coil and holding register values in a manner similar to a real PLC. For each HMI–PLC pair, Modbus network traffic will be captured by the tcpdump utility. This captured traffic is used to check if packet loss or network errors occur in the virtualized hosts or network during the experiment.



**Fig. 3 Field network VMs**

The HMI and PLC VMs are hosted by a VMware ESXi hypervisor on a Dell R610 server. For each HMI–PLC pair, a virtual switch connects the 2 VMs. Each virtual switch is part of the same virtual network. The virtual network is also mapped to one of the host machine’s network interface cards (NICs) and this NIC allows external access to the virtual network, for example, to the attacker.

### 2.3 PLC Configuration

Each ModbusPal virtual PLC instance must be configured with a set number of holding registers and coils to simulate the corresponding process presented in Figs. 1 and 2. ModbusPal was configured with an Extensible Markup Language (XML)-based text file where holding registers and coils are defined and values specified. The values of holding registers and coils can be controlled programmatically within ModbusPal.

In Appendix B. ModbusPal Tables, we list the detailed configuration information for each of the 6 PLCs controlling the 6 processes (see Fig. 2):

1. Chicken cooker

2. Vegetable cooker
3. Meal preparation and packaging
4. High-pressure processing
5. Main conveyor belt
6. Packaging

## **2.4 HMI Configuration**

---

Each virtualized Mango HMI polls its respective ModbusPal PLC for its values of coils and holding registers every 10 seconds (sec). The Mango software will send Modbus TCP requests to ModbusPal to request values of all holding registers and coils configured for this experiment. A graphical dashboard will also be configured to provide situational awareness, see Fig. 4 for the overall dashboard, which represents the view typically seen in industrial plants.

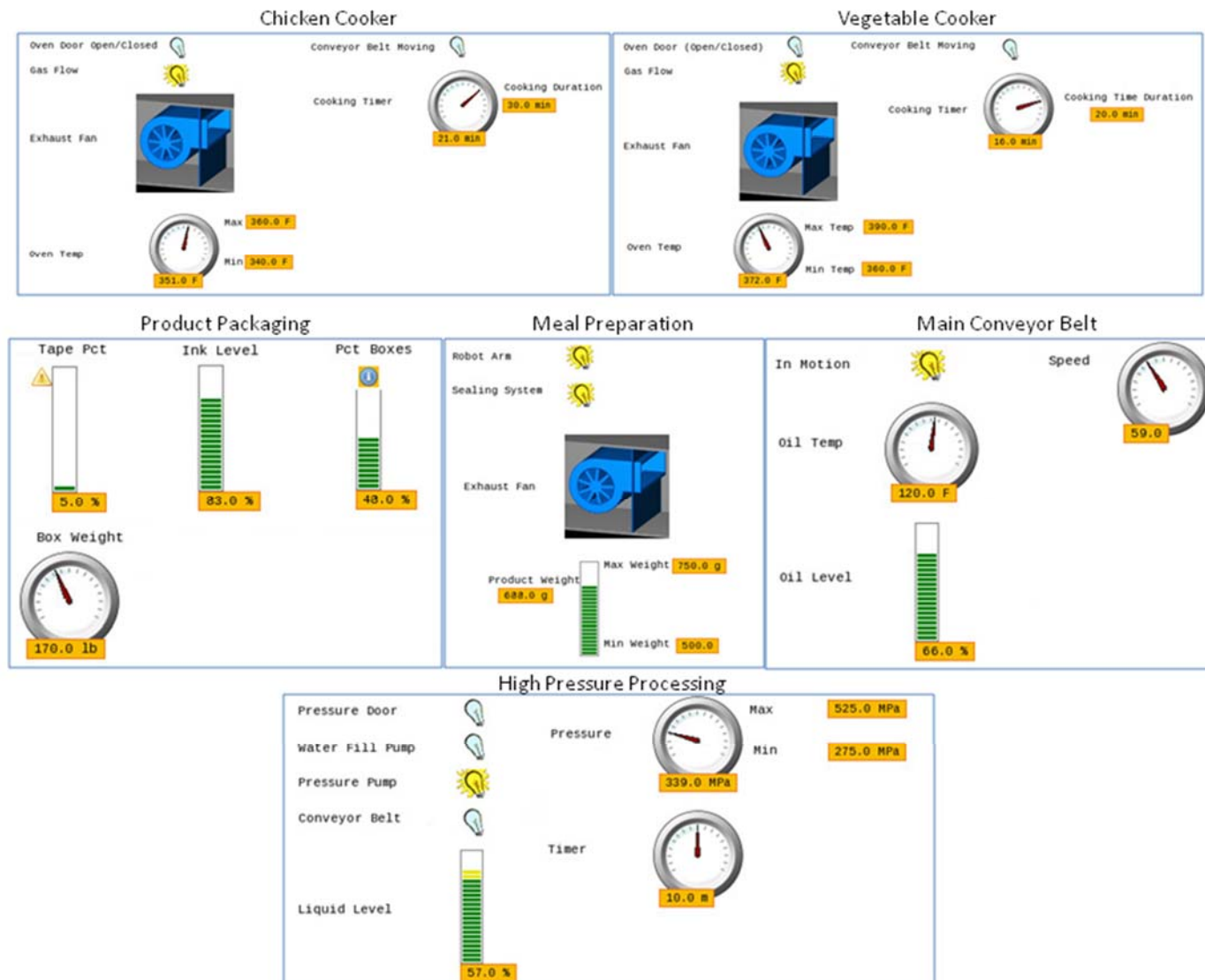
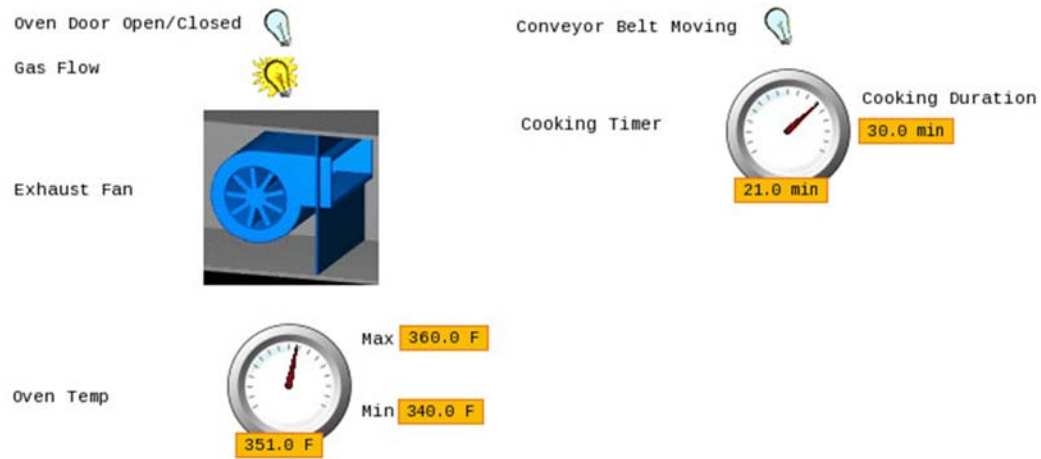
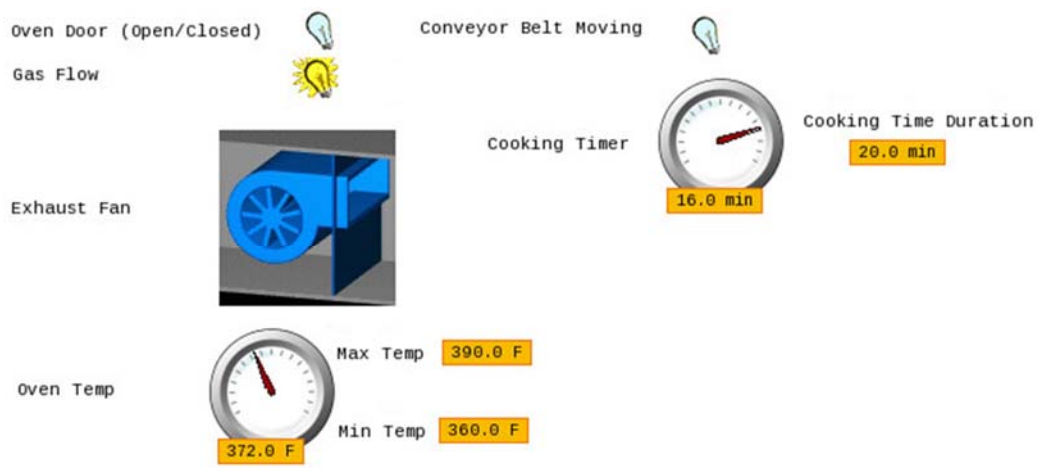


Fig. 4 Overall plant HMI dashboard

Snapshots are shown of the 6 Mango HMIs in Figs. 5–10, illustrating the HMI dashboards of each of the 6 processes.



**Fig. 5 Chicken cooker dashboard**



**Fig. 6 Vegetable cooker dashboard**

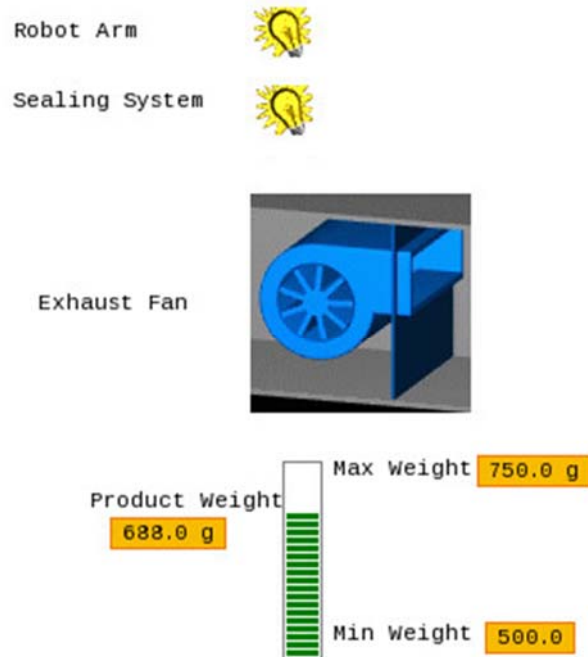


Fig. 7 Meal preparation dashboard

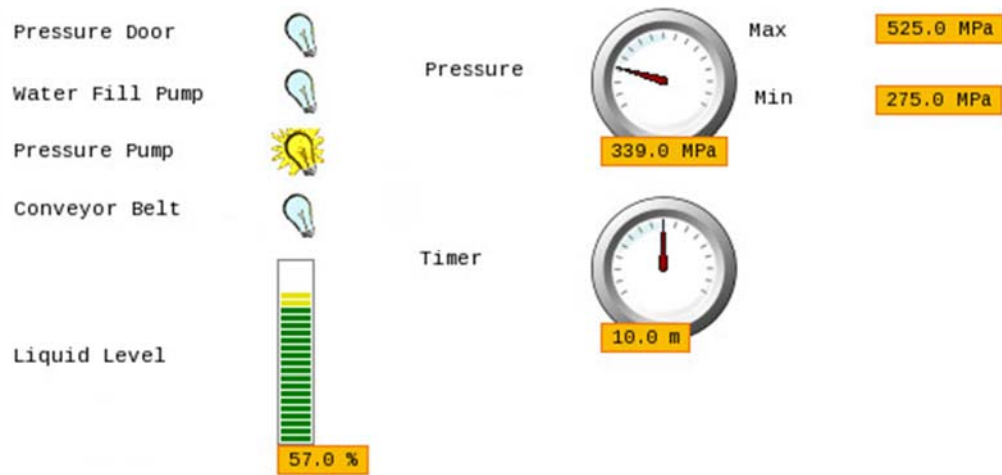


Fig. 8 High-pressure processing dashboard



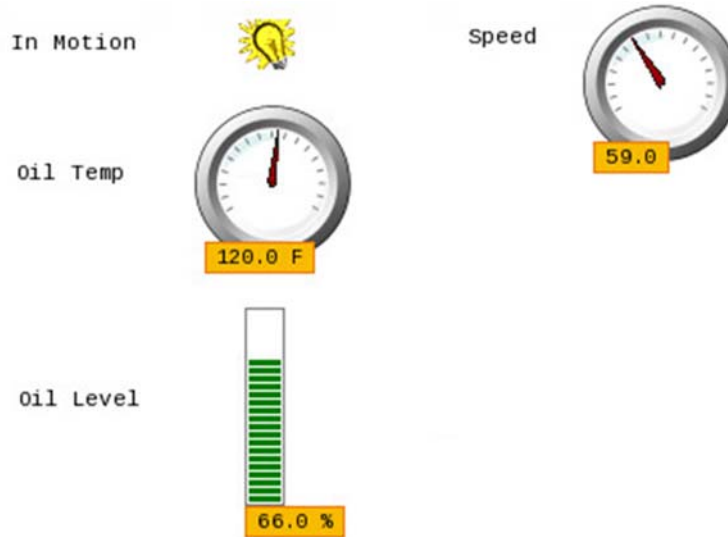


Fig. 9 Main conveyor belt dashboard

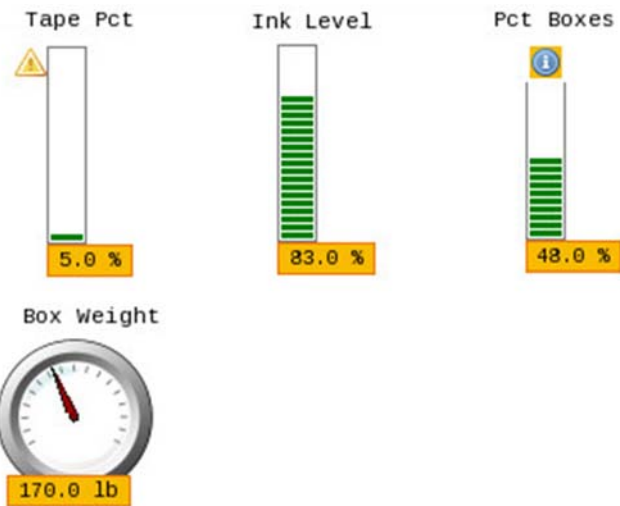


Fig. 10 Product packaging dashboard

### 3. Execution of MRE Test

---

The MRE test consists of 2 parts—a network virtualization test and a simulated cyber-attack. A description of these 2 subtests follows.

#### 3.1 Network Virtualization Subtest

---

In this subtest, we will validate that each HMI-PLC pair of VMs has network connectivity and that the network paths are configured correctly.

Step 1: For each automation process, once the Mango HMI begins polling its respective ModbusPal instance, capture the traffic over a 24-hour (h) period using tcpdump.

Step 2: During the 24-h polling process, perform spot checks to verify the Mango HMI is receiving measurements in compliance with the values listed in Tables 2–7.

Step 3: Use Wireshark to inspect the 24-h tcpdump captures and check for any Internet Control Message Protocol (ICMP) error messages in the tcpdump files. The condition for PASS requires that no ICMP error messages exist in the tcpdump files. The condition for FAIL requires that one or more ICMP error messages are found. If ICMP error messages in the tcpdump files are discovered, investigate the reasons and correct the configuration.

### **3.2 Simulated Cyber Attack**

---

This subtest simulates a cyber-attacker sending malicious Modbus messages to a PLC to change the values of coils. The Modbus protocol does not have security capabilities to authenticate messages or prevent replay attacks.<sup>2</sup> As a result, anyone (insider or external threat actor) who has knowledge of the process map can send malicious Modbus messages to a PLC and impact an automation process. External threat actors can gain knowledge of the process map and PLC ladder logic by conducting reconnaissance of the plant network prior to an attack.

In this subtest, we will conduct a manipulation of view attack on the Meal Preparation ModbusPal instance. In a real plant environment, this attack would cause the production process to stop while plant operators investigate the cause.

Step 1: On an external laptop connected to the experiment network, use the Perl “mbtget” script to change the Meal Preparation PLC Robot Arm and Sealing System coil values to “0” (“Off” state).

Step 2: Monitor the Meal Preparation process Mango HMI dashboard. The dashboard should show the Robot Arm and Sealing System processes are in an “Off” state and an alarm should be visible. The test is a PASS if the Meal Preparation dashboard shows both processes are “Off” and alarm symbols are displayed.

## 4. MRE Test Results

---

### 4.1 Network Virtualization Subtest

---

The tcpdump data captured over a 24-h period of each HMI-simulated PLC pair were examined using Wireshark. The table presents the number of packets captured and examined for each HMI and simulated PLC pair. The number of network errors are also listed. Because no network errors were found, all tests were a PASS.

**Table Network virtualization test results**

Automation Process	Number of Mango HMI– ModbusPal Packets Captured over 24 h	Number of Network Errors	Test Results (PASS/FAIL)
Chicken cooker	172,904	0	PASS
Vegetable cooker	172,806	0	PASS
Meal preparation	172,818	0	PASS
High-pressure processing	172,807	0	PASS
Main conveyor belt	172,803	0	PASS
Product packaging	86,402	0	PASS

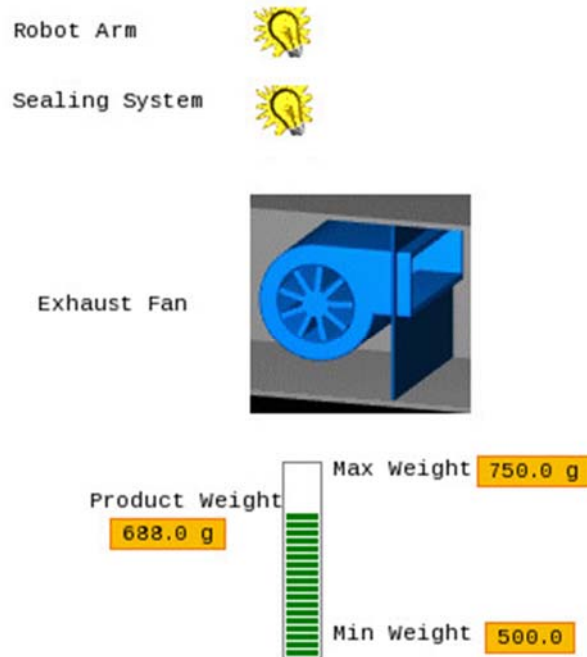
For each automation process, the Mango HMI polled its respective ModbusPal application every 10 sec. In each polling period, Mango HMI issued a Modbus coil read request and waited for the response. After receiving the coil measurements, the Mango HMI sent a holding register read request to its respective ModbusPal application. Therefore in each 10-sec poll interval, 2 Modbus read requests are sent and 2 responses are received by the HMI.

The number of Modbus packets for the Product Packaging process was much less than the other automation processes because Product Packaging only used holding registers. Therefore, in each 10-sec polling interval, Mango sent only one Modbus message compared to 2 in the other automation processes.

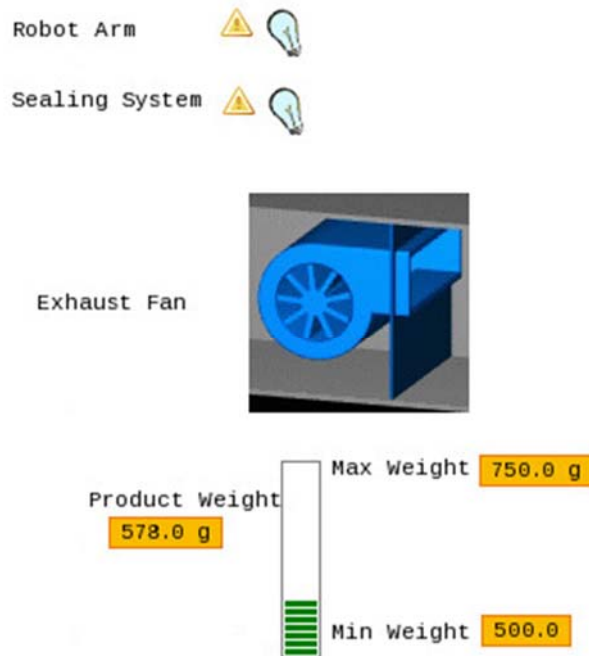
### 4.2 Simulated Cyber Attack

---

We show the Meal Preparation HMI dashboard during normal operations and after the attacker has sent malicious traffic, in Figs. 11 and 12, respectively.



**Fig. 11** Meal preparation dashboard before cyber attack

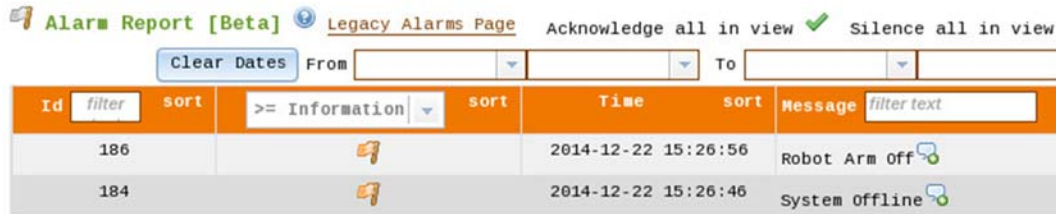


**Fig. 12** Meal preparation dashboard after cyber attack

The attack was simulated using the Perl mbtget script, which sent Modbus coil write messages to the Meal Preparation ModbusPal to set the coil values to “0” (turn the process offline). The small yellow triangles with an “!” symbol in the

upper part of Fig. 12 are alarms that have consequently sounded in the Mango dashboard for this critical process.

Fig. 13 is a screen capture of the Meal Preparation HMI alarm panel. The loss of the Robot Arm and Sealing System processes are listed as critical alarms.



The screenshot shows the 'Alarm Report [Beta]' interface. At the top, there are links for 'Legacy Alarms Page', 'Acknowledge all in view' (with a green checkmark), and 'Silence all in view'. Below these are date filters: 'Clear Dates', 'From', and 'To'. The main table has columns for 'Id', 'filter', 'sort', '>= Information', 'sort', 'Time', 'sort', and 'Message filter text'. Two alarms are listed:

Id	filter	sort	>= Information	sort	Time	sort	Message	filter text
186					2014-12-22 15:26:56		Robot Arm Off	
184					2014-12-22 15:26:46		System Offline	

**Fig. 13 Meal preparation alarm panel after cyber attack**

This attack would have resulted in a shutdown of the Meal Preparation process if this were an actual plant. The test result is PASS.

## 5. Conclusions

This experiment demonstrates that virtualization of SCADA components is an effective means to simulate a production plant's network traffic and create cyber-attack scenarios. The VMs and guest operating systems with their applications emulated the automation components found in a plant and zero packets were lost by the virtual network. The virtual environment enabled us to simulate a cyber-attack on a commonly used Modbus industrial protocol. We will leverage the results of this experiment in future tests to protect critical infrastructure.

## 6. References

---

1. Mango Automation. Version 2.4.2, Intelligent Automation Systems, Inc.  
<http://infiniteautomation.com/index.php/software/>. [accessed Oct 8 2014].
2. Modbus. Digital Bond Incorporated. n.d.  
<http://www.digitalbond.com/scadapedia/protocols/modbus-2/>. [accessed Jan 14 2015].

## **Appendix A. Experiment Hardware and Software**

---

Table A-1 presents each hardware component with a description of its use and operating system.

**Table A-1 Hardware list**

Platform	Function	Operating System
Mac laptop and desktop	Remote access to Virtual Machines (VMs), configure applications for experiment	OS X Mavericks (Version 10.9)
Dell R610	Hosts ESXi	ESXi 5.5 hypervisor
Dell R710	Software development and testing of applications	CentOS 6.5
Dell PowerConnect 6224	Network switch	Dell Firmware

The software for this experiment is presented in Table A-2 for each hardware platform. This experiment will use US Army Research Laboratory (ARL) licensed, as well as open source software and operating systems.

**Table A-2 Software list**

Software	Function	Platform
VirtualBox	Hosts Windows Vista on Mac platforms	Mac laptop and desktop computers
Windows Vista Enterprise	Guest OS of VirtualBox. Enables Mac users to access ESXi VMs using vSphere client.	Mac laptop and desktop computers
ESXi 5.5	Hypervisor to host guest VMs	Dell R610
vSphere Client 5.5	Remote access to ESXi VMs	Mac laptop and desktop computers
CentOS 6.5	Operating system	Dell R710 and each VM hosting the simulated PLC and Mango HMI
Java Software Development Kit (JDK) 1.7	Compile ModbusPal PLC simulator	Dell R710
Java Runtime Environment (JRE) 1.7	Run Mango HMI and ModbusPal PLC simulator	Dell R710 and each VM
Eclipse	Development tool to program the ModbusPal PLC simulator	Dell R710
Perl 5.10.1	Runs mbtget script to simulate a cyber attacker	Cyber attacker VM, Dell R710
Mango	HMI which polls simulated PLC (ModbusPal) for status messages	VMs simulating an HMI workstation
ModbusPal	Simulates a PLC	VMs simulating a PLC
mbtget	Simulates a cyber attacker. Sends scripted Modbus messages to simulated PLCs.	Cyber attacker VM
tcpdump	Captures Modbus packets	Each VM



## **Appendix B. ModbusPal Tables**

---

Tables B-1 through B-6 list the coil and holding register configuration of each ModbusPal application to emulate its respective Programmable Logic Controller (PLC) depicted in the Fig. 1 process map.

**Table B-1 Configuration and measurements for chicken cooker PLC**

Entity	Holding Register Index	Coil Index	Allowed Value	Value Set for Experiment	Data Type
Oven Door Open/Closed	...	1	1 = Oven door is OPEN 0 = Oven door is Closed	0	Bit
Gas Flow On/Off	...	2	1 = Gas turned ON to oven 0 = Gas turned OFF to Oven	1	Bit
Exhaust Fan On/Off	...	3	1 = Exhaust Fan is ON 0 = Exhaust Fan is OFF	1	Bit
Conveyor In Motion	...	4	1 = belt is moving forward 0 = belt is stopped	0	Bit
Oven Temperature (°F)	1	...	345–355 when Oven is ON	ModbusPal automation to randomly choose values between 345–355	2 Byte Signed Integer
Oven Temperature Maximum (°F)	2	...	360	360	2 Byte Signed Integer
Oven Temperature Minimum (°F)	3	...	340	340	2 Byte Signed Integer
Cooking Time Remaining (min)	4	...	0–30	ModbusPal automation to linearly decrement time from 30 to 0	2 Byte Signed Integer

**Table B-2 Configuration and measurements for vegetable cooker PLC**

Entity	Holding Register Index	Coil Index	Allowed Values	Value Set for Experiment	Data Type
Oven Door Open/Closed	...	1	1 = Oven door is OPEN 0 = Oven door is Closed	0	Bit
Gas Flow On/Off	...	2	1 = Gas turned ON to oven 0 = Gas turned OFF to Oven	1	Bit
Exhaust Fan On/Off	...	3	1 = Exhaust Fan is ON 0 = Exhaust Fan is OFF	1	Bit
Conveyor In Motion	...	4	1 = Belt is moving forward 0 = Belt is stopped	0	Bit
Oven Temperature (°F)	1	...	370–380 when Oven is ON	ModbusPal automation to randomly choose values between 370–380	2 Byte Signed Integer
Oven Maximum Temperature Alarm Set Point (°F)	2	...	390	390	2 Byte Signed Integer
Oven Minimum Temperature Alarm Set Point (°F)	3	...	360	360	2 Byte Signed Integer
Cooking Time Remaining (min)	4	...	0–20	ModbusPal automation to linearly decrement time from 20 to 0	2 Byte Signed Integer
Cooking Time Duration (min)	5	...	20	20	2 Byte Signed Integer

**Table B-3 Configuration and measurements for meal preparation and packaging PLC**

Entity	Holding Register Index	Coil Index	Allowed Values	Value Set for Experiment	Data Type
Robot Arm Online	...	1	1 = Robot Arm is in operation 0 = Robot Arm is offline	1	Bit
Sealing System Online	...	2	1 = Sealing System is on operation 0 = Sealing System is offline	1	Bit
Exhaust Fan On/Off	...	3	1 = Exhaust Fan is ON 0 = Exhaust Fan is OFF	1	Bit
Product Weight (grams)	1	...	510–740 grams	ModbusPal automation to randomly choose values between 510–740 grams	2 Byte Signed Integer
Product Weight Maximum Alarm Set Point (grams)	2	...	750	750	2 Byte Signed Integer
Product Weight Minimum Alarm Set Point (grams)	3	...	500	500	2 Byte Signed Integer

**Table B-4 Configuration and measurements for high-pressure processing PLC**

Entity	Holding Register Index	Coil Index	Allowed Value	Value Set for Experiment	Data Type
Pressure Door Open/Closed	...	1	1 = Pressure door is OPEN 0 = Pressure door is Closed	0	Bit
Water Fill Pump On/Off	...	2	1 = Water Fill Pump is ON 0 = Water Fill Pump is OFF	0	Bit
Pressure Pump On/Off	...	3	1 = Pressure Pump is ON 0 = Pressure Pump is OFF	1	Bit
Product Pressuring Process On/Off	...	4	1 = Product is being pressurized 0 = Product is not being pressurized	1	Bit
Conveyor Belt In Motion	...	5	1 = Belt is moving forward 0 = Belt is stopped	0	Bit
Liquid Level Percent (%) Full	1	...	40–60 % when products are being pressurized	ModbusPal automation to randomly choose values between 40–60	2 Byte Signed Integer
Pressure (MPa)	2	...	300–500 when pressurizing process is ON	ModbusPal automation to randomly choose values between 300–500	2 Byte Signed Integer
Maximum Pressure Alarm Set Point (MPa)	3	...	275	275	2 Byte Signed Integer
Minimum Pressure Alarm Set Point (MPa)	4	...	525	525	2 Byte Signed Integer
Pressuring Time Remaining (s)	5	...	0–20	ModbusPal automation to linearly decrement time from 20 to 0	2 Byte Signed Integer

**Table B-5 Configuration and measurements for main conveyor belt PLC**

Entity	Holding Register Index	Coil Index	Allowed Value	Value Set for Experiment	Data Type
Conveyor Belt In Motion	...	1	1 = Belt is moving 0 = Belt is stopped	1	Bit
Motor Oil Temperature	1		80–150 °F	ModbusPal automation to randomly choose values between 80–150	2 Byte Signed Integer
Motor Oil Level (% Full)	2	...	45–70% full	ModbusPal automation to randomly choose values between 45–70	2 Byte Signed Integer
Speed (surface feet per minute [FPM])	3	...	55–70 FPM when the conveyor belt is moving	ModbusPal automation to randomly choose values between 55–70	2 Byte Signed Integer

**Table B-6 Configuration and measurements for packaging PLC**

Entity	Holding Register Index	Coil Index	Allowed Value	Value Set for Experiment	Data Type
Packing Tape Available (% Full)	1	...	0–100%	ModbusPal automation to linearly decrease percentage from 100 to 0	2 Byte Signed Integer
Ink Level (%)	2	...	0–100%	ModbusPal automation to linearly decrease percentage from 100 to 0	2 Byte Signed Integer
Shipping Container Weight (lbs)	3	...	150–200 lbs	Used ModbusPal automation to randomly set value between 150 to 200	2 Byte Signed Integer
Shipping Box Inventory (%)	4	...	0–100%	ModbusPal automation to linearly decrease percentage from 100 to 0.	2 Byte Signed Integer

## List of Symbols, Abbreviations, and Acronyms

---

ACAL	US Army Cyber Analytics Laboratory
ARL	US Army Research Laboratory
FPM	feet per minute
h	hour
HMI	human machine interface
ICMP	Internet Control Message Protocol
ICS	Industrial Control Systems
JDK	Java Development Kit
JRE	Java Runtime Environment
Mac	Macintosh
MRE	Meals-Ready-to-Eat
NIC	network interface card
PLC	programmable logic controller
OS	operating system
SBNAB	Sustaining Base Network Assurance Branch
SCADA	Supervisory Control and Data Acquisition
sec	second(s)
TCP	transmission control protocol
VM	Virtual Machine
XML	Extensible Markup Language

1 DEFENSE TECHNICAL  
(PDF) INFORMATION CTR  
DTIC OCA

1 DIRECTOR  
(PDF) US ARMY RESEARCH LAB  
RDRL CIO LL  
IMAL HRA MAIL & RECORDS MGMT

3 DIRECTOR  
(PDF) US ARMY RESEARCH LAB  
RDRL CIN S  
D SULLIVAN  
E COLBERT  
R RESCHLEY